Interim Report

Jung Lin (Doris) Lee

September 7, 2014

Group: Jennifer Ito, Manuel Silva

1 First Set of Data (lower intensity; N = 10,000)

```
In [116]: analysis("msjidlmh_140906_1051_40.csv")
```

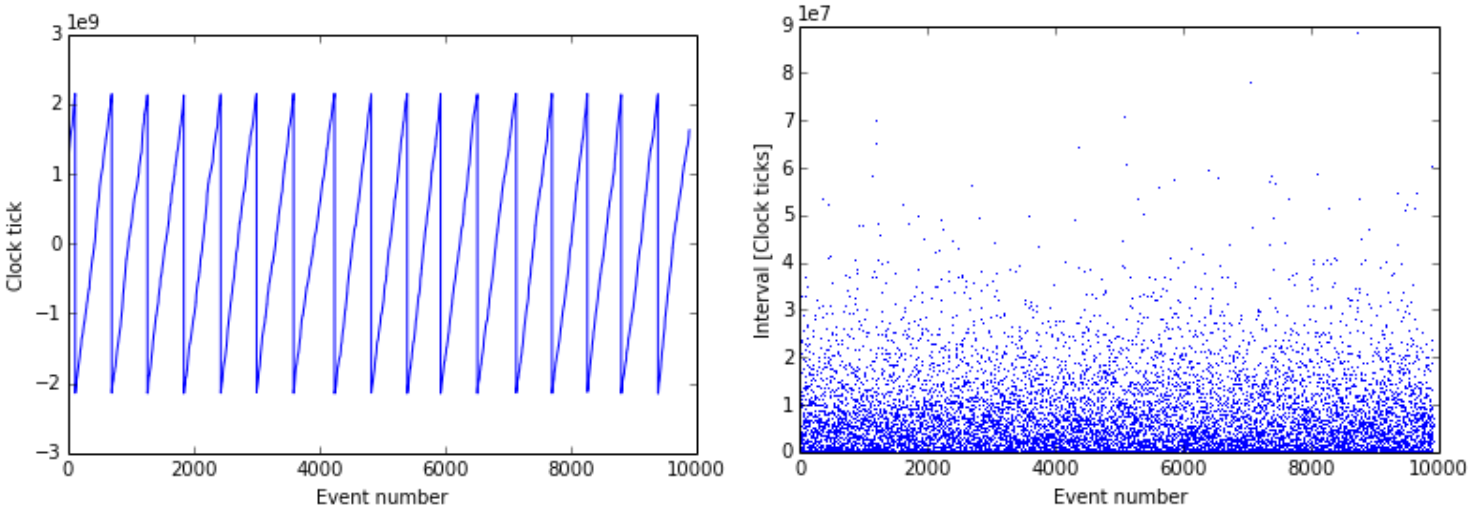


Figure 1: Clock tick plot showing the event cycle, the max and min correspond to maximum signed integer stored on a 32bit program(left); Event Scatterplot (right)

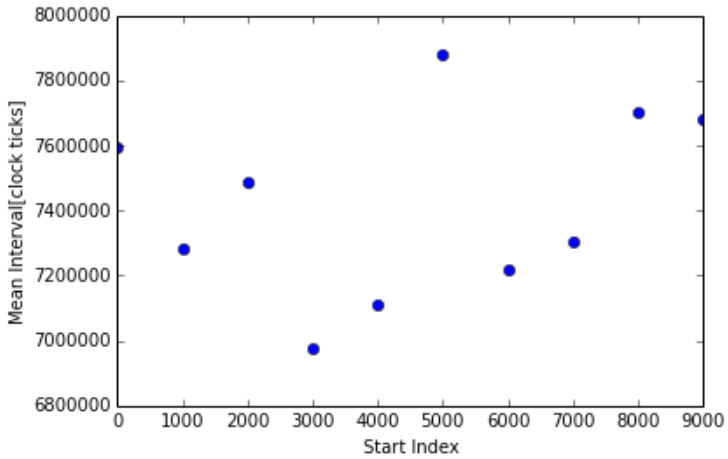


Figure 2: Taking 10 different lengths of data slices, each with steps of 1000. N=10,000

2 Second Set of Data(max intensity; N = 10,000)

This makes sense because there is the same number of event as dataset 1 but they have higher intensity so there are more photons coming at the same time. Therefore it only took ˜2 clock cycles for the 10,000 photon count to run out.

```
In [117]: analysis("msjidlmh_140906_1052_40.csv")
```

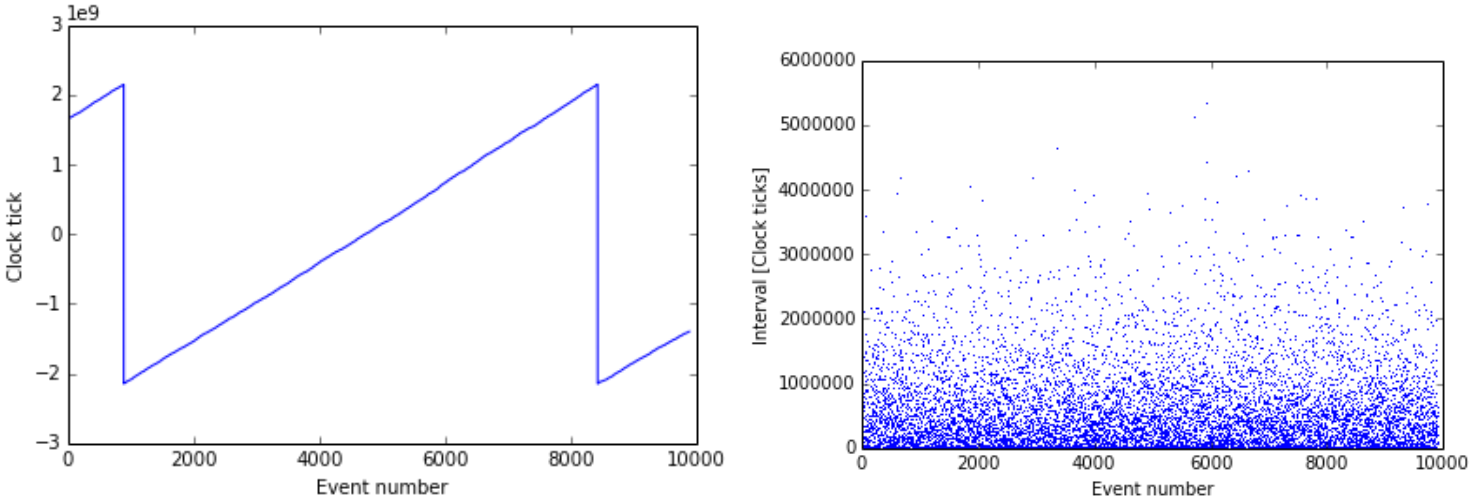


Figure 3: Clock tick plot (left); Event Scatterplot (right)

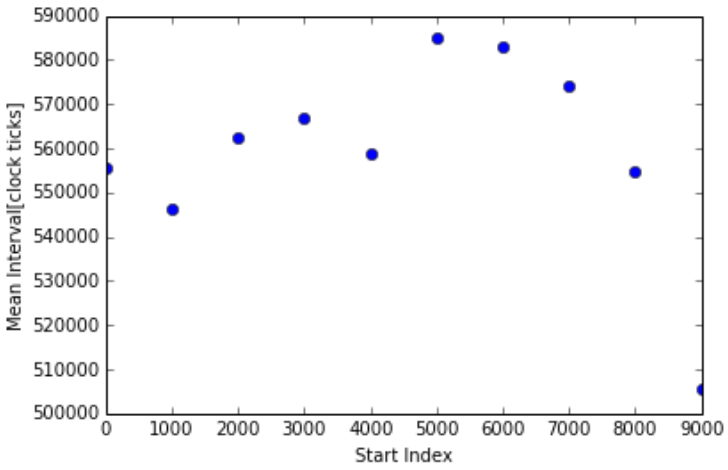
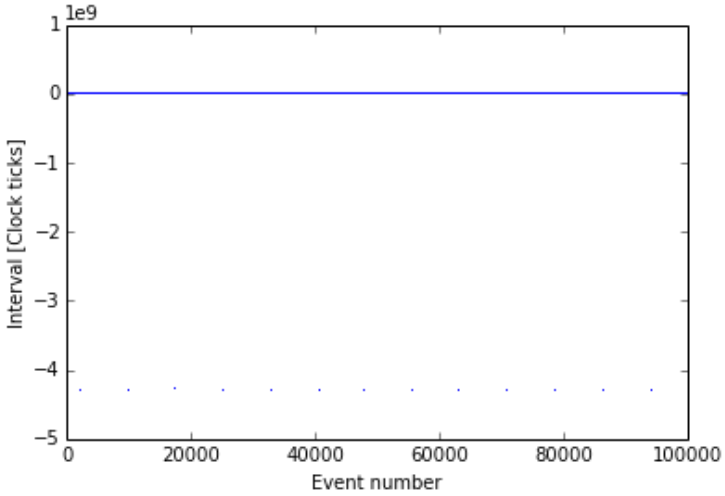


Figure 4: Taking 10 different lengths of data slices, each with steps of 1000. N=10,000

3 Third set of data(max intensity; N = 100,000)

```
In [118]: analysis("msjidlmh_140906_1057_50.csv")
```

When `dtype=int32` is left out, the datalist assumes the default return type of `loadtxt` (floating point). Since floating point numbers are stored with limited precision in Python, it is forced to approximate the integer as a float. As a result, the last few digits of the integer may be off. Since the values of two neighboring data points we are subtracting to obtain `dt` usually lie very close since they occur right after another, therefore errors introduced by the floating point approximation results in a visible difference when we are trying to compare its difference with a subsequent datapoint as shown in the plot below.



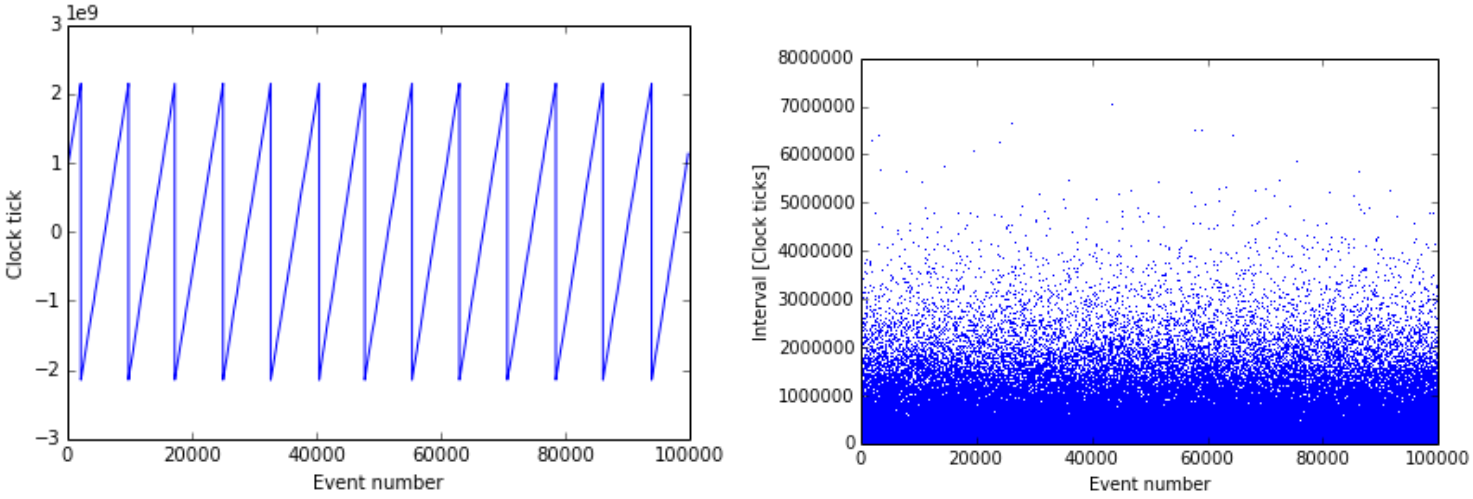


Figure 5: Clock tick plot (left); Event Scatterplot (right)

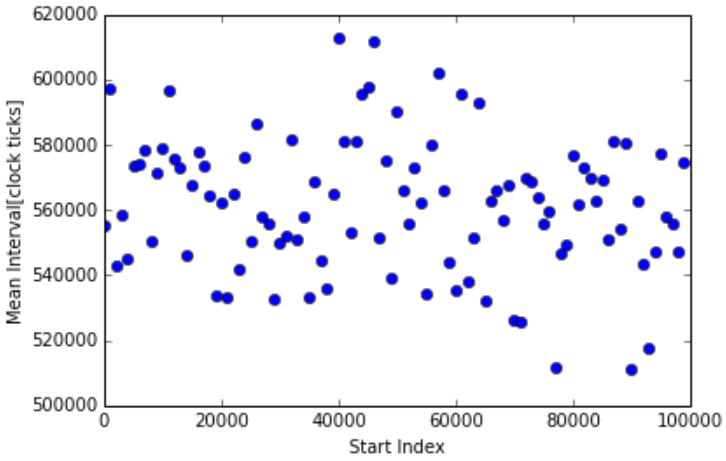
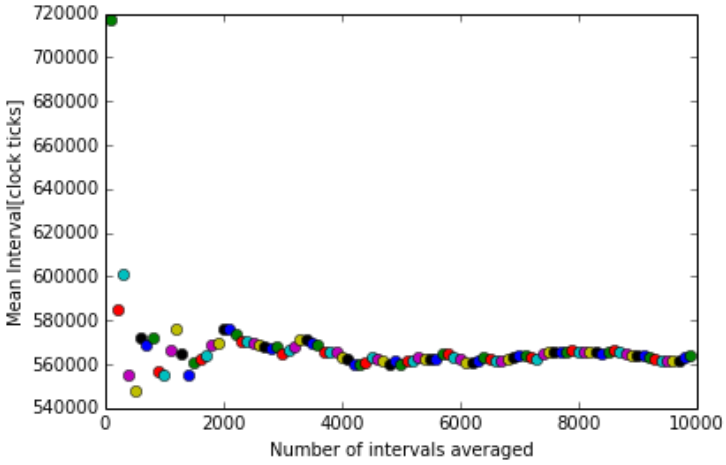


Figure 6: Taking 100 different lengths of data slices, each with steps of 1000. N=100,000

4 Statistics

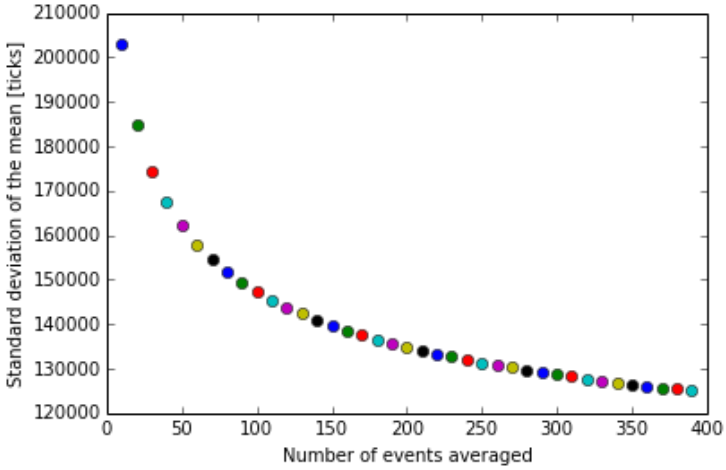
Most of the statistical analysis for sec.3 is done using dataset#3 since it is there is a greater number of sample. (N is large) As shown in the figure below, when we take greater number of datapoints in the averaged interval, the value of mean interval approaches a limit. It converges onto the mean μ.



4.1 Standard deviation of the mean

Looping through `nstep` so that we can change the size of the dataset chunk and compute the mean for each. Then use the mean that we found as the average so that values in `marr` has something to compare against:

$$s = \sqrt{\frac{\sum_{i=1}^{N}(\mu_i - \mu)^2}{N-1}}$$



Here we plot the same variables but the x axis is scaled by $1/\sqrt{N}$ since the standard deviation of the mean = s/$\sqrt{N}$, the resulting graph is approximately linear.